

An oil painters recognition method based on cluster multiple kernel learning algorithm

Liao, Zhifang; Gao, Le; Zhou, Tian; Fan, Xiaoping; Zhang, Yan; Wu, Jinsong

Published in:
IEEE Access

DOI:
[10.1109/ACCESS.2019.2899389](https://doi.org/10.1109/ACCESS.2019.2899389)

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

[Link to publication in ResearchOnline](#)

Citation for published version (Harvard):
Liao, Z, Gao, L, Zhou, T, Fan, X, Zhang, Y & Wu, J 2019, 'An oil painters recognition method based on cluster multiple kernel learning algorithm', *IEEE Access*, vol. 7, pp. 26842-26854.
<https://doi.org/10.1109/ACCESS.2019.2899389>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please view our takedown policy at <https://edshare.gcu.ac.uk/id/eprint/5179> for details of how to contact us.

Received January 25, 2019, accepted February 8, 2019, date of publication February 14, 2019, date of current version March 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2899389

An Oil Painters Recognition Method Based on Cluster Multiple Kernel Learning Algorithm

ZHIFANG LIAO¹, LE GAO¹, TIAN ZHOU², XIAOPING FAN³,
YAN ZHANG⁴, AND JINSONG WU⁵

¹School of Computer Science and Engineering, Central South University, Changsha 410075, China

²School of Information Science and Engineering, Central South University, Changsha 410075, China

³School of Information Technology and Management, Hunan University of Finance and Economics, Changsha 410205, China

⁴Department of Computer, Communication and Interactive System, School of Engineering and Built Environment, Glasgow Caledonian University, Glasgow G4 0BA, U.K.

⁵Department of Electrical Engineering, Universidad de Chile, Santiago 8370451, Chile

Corresponding author: Xiaoping Fan (xpfan@csu.edu.cn)

This work was supported in part by the Key Laboratory Funding under Grant 614250304010617, in part by the Hunan Provincial Key Laboratory of Finance and Economics Big Data Science and Technology (Hunan University of Finance and Economics) under Grant 2017TP1025, and in part by the HNSF under Grant 2018JJ2535.

ABSTRACT A lot of image processing research works focus on natural images, such as in classification, clustering, and the research on the recognition of artworks (such as oil paintings), from feature extraction to classifier design, is relatively few. This paper focuses on oil painter recognition and tries to find the mobile application to recognize the painter. This paper proposes a cluster multiple kernel learning algorithm, which extracts oil painting features from three aspects: color, texture, and spatial layout, and generates multiple candidate kernels with different kernel functions. With the results of clustering numerous candidate kernels, we selected the sub-kernels with better classification performance, and use the traditional multiple kernel learning algorithm to carry out the multi-feature fusion classification. The algorithm achieves a better result on the Painting91 than using traditional multiple kernel learning directly.

INDEX TERMS Oil painters recognition, multiple kernel learning.

I. INTRODUCTION

In the long history of human development, art plays a very important role. As an important form of culture and art, art works represent a unique and significant way for the human to observe and express the world, for which not only reflects the artists' superb skills and profound ideas, but also shows a nation's cultural landscape and aesthetic taste.

The study of art painting is not only limited to the field of art. With the combination of computer and art, the analysis of large-scale painting images has become an important research interest of computer vision [1]. Generally, painting works usually show the artistic style, feature, manner, taste and artistic air, that is, the relatively stable and overall art features displayed by the interactive effects between an artist's creative style and the languages and semantic factors. The identification and recognition of the different paintings' painters is an important means to classify paintings for art researchers.

The associate editor coordinating the review of this manuscript and approving it for publication was Wenbing Zhao.

With the rapid development of computer vision in recent years, how to use the computer to recognize, and more, how to use mobile device to recognize and identify the authorship of paintings is a research focus in the field of computer vision [2]. Most of the research about traditional computer image classification focus on the classification and recognition of natural images. Strictly speaking, the research field is of the objective classification, which means the judgment of target object is objective. Since a painter has his own subjectivity in the creation, the classification of paintings differs greatly from that of natural images. Therefore, it is the point of how to accurately describe art features of various painters with mathematical models effectively, to design a robust algorithm to analyze painting attributes of different painters and to classify various paintings. In this paper, the cluster multiple kernel learning (CMKL) algorithm is used, which avoids the problem of redundant solution and high computational complexity produced by traditional multiple kernel learning (MKL) algorithm [3]. The rest of this paper is laid out as follows:

Section 1 introduces the research background and significance of the thesis, presenting the relevant overview about the classification of oil painters and the structure of the paper.

Section 2 introduces the current situation about recognition of oil painters and multiple kernel learning both at home and abroad.

Section 3 describes the global features and local features extracted from oil paintings.

Section 4 elaborates the theoretical derivation and complexity analysis of the CMKL algorithm.

Section 5 verifies the validity of the CMKL algorithm through single classifier and multi-class classifier experiments. And it also provides a mobile application process for oil painter recognition.

Section 6 concludes the CMKL algorithm proposed in this paper and gives the further re-search direction.

II. RELATED WORK

A. THE STATUS OF RESEARCH ON OIL PAINTINGS RECOGNITION

Oil painting, created by the Dutch in the 14th century and popular in the European countries, as a part of religious activities and in the service of the religion, is the main way of painting in Western countries. In 2003, Widjaja *et al.* [4] reviewed the application of image processing technology in art, explored various color models used in representing color features, compared the performance of several versions of multi-category support vector machines for oil paintings classification, and finally found out that several weighted combination of oriented acyclic graph SVM and Gaussian kernel performs best for classification. In 2004, Lyu *et al.* [5] studied the digital technique for art authentication, that is, building a statistical model of an artist from the scans of a set of authenticated works. The statistical model consists of first and higher-order wavelet statistics. In 2008, Johnson *et al.* [6] used the 2-D hidden Markov model and wavelet transform to extract the brushstroke features automatically from the oil paintings so as to study the artist's painting style; In 2009, Shen [7] achieved the classification of the oil painters by retrieving the local and global features from the images and using radial basis function neural network as the base classifier for integrated learning; Zujovic *et al.* [8] proposed a classification algorithm by artistic genre based on salient features of images by exploiting the way humans perceive a painting; Siddiquie *et al.* [9] proposed a discriminative kernel-based method, where training data instances are selected using AdaBoost, performing very well in classifying natural images and oil paintings. In 2010, Shamir *et al.* [10] used a large set of image features and image transforms to analyze and employed Fisher scores to assess the computed image descriptors, and the most informative features were used for the classification and similarity measurements of paintings, painters, and schools of art; In 2012, Carneiro *et al.* [11] presented a new database of monochromatic artistic images for artistic image understanding, which indicates that the

automatic classification of digital paintings has become an open research field. In 2014, Khan *et al.* [12] produced the largest scale data set at that time, consisting of 4266 works from 91 different artists. By combining different manual features, and then using SVM classifier to learn, they succeeded in painters identification and styles classification; In 2015, Gatys *et al.* [13] achieved synthesis of the content from a common picture and the style of world-famous paintings by using Deep Neural Network VGG, thus ordinary pictures have a sense of famous artistic paintings. Peng and Chen [14] classified oil paintings by extracting deep features from multiple layers of convolutional neural networks. Peng and Chen [15] proved that, in addition to the classification of paintings, deeper convolution layers can produce excellent results for material and texture recognition. In 2016, Folego *et al.* [16] extracted features from the local area of images by using convolutional layers of the deep neural network VGG, used a number of SVM classifiers to learn the local features, and finally used the method of decision fusion to identify paintings of Van Gogh; Anwer *et al.* [17] proposed an approach that based on VGG 16 and Deformable Part Model (DPM) which can extract features from one painting at the same time, and then they fused the features extracted by VGG 16 and DPM through multi-scale activation layer and Fisher Vector overlay. Ultimately they used linear SVM to classify paintings.

B. RESEARCH STATUS OF MULTIPLE KERNEL LEARNING

In 2004, Lanckriet *et al.* [18] obtained the kernel matrix from data sets by using Semi-Definite Programming (SDP) technique, and gave an effective transductive algorithm. Subsequently, Bach *et al.* [19] proposed a novel dual formulation of the QCQP (Quadratically-Constrained Quadratic Program) as a second-order cone programming, SOCP) and model a more effective algorithm based on SMO (Sequential Minimal Optimization) techniques to yield a formulation by exploiting the technique of Moreau-Yosida regularization [20]. Rakotomamonjy *et al.* [21] proposed a novel MKL algorithm called SimpleMKL and demonstrated the equivalence between SimpleMKL and MKL algorithm based on mixed-norm regularization. Kloft *et al.* [22] proposed a new non-sparse approach to MKL for some specific non-sparse application problems by changing the traditional L1-norm constraints to L2-norm constraints. Then, Kloft *et al.* [23] generalized MKL to arbitrary L_p norms ($p > 1$) and devised new insights on the connection between several existing MKL formulations, designed two efficient alternative optimization strategies and developed two efficient interleaved optimization strategies for arbitrary $p > 1$. In 2010, Xu *et al.* [24] formulated a closed-form solution for optimizing the kernel weights based on the equivalence between group-lasso [25] and MKL and generalized to the case for L_p -MKL ($p > 1$) regularization. Xu *et al.* [26] proposed a novel Soft Margin framework and demonstrated that many of the existing MKL algorithms can be viewed as special cases under their Soft Margin framework,

and designed effective optimization methods. Therefore, in practical application multiple kernel learning with better learning ability is used more widely than single kernel learning, providing a new idea for kernel method to be applied in solving complicated practical problems.

This paper proposes a cluster multiple kernel learning (CMKL). Before multiple kernel learning is carried out, a pre-learning process is used to cluster similar candidate sub-kernels and select some sub-kernels with better classification ability, which decreases the size of candidate sub-kernels and reduces the computational complexity of multiple kernel learning.

III. EXTRACTION OF OIL PAINTINGS' FEATURES

Most of the computer vision research mainly focuses on the analysis of the image contents, such as the shape and the key points, etc., while the artistic style of the image is a high-level semantic domain which is hard to describe comprehensively with the basic features only. Therefore, proceeding from the painting techniques, this paper extracts global features and local features based on Bag-of-Features from color, texture and spatial layout. In order to get the same scale image features, we scale each input image to 128×128 .

A. EXTRACTION OF GLOBAL FEATURE

The global feature incorporates the characteristics of the entire image and it describes an image as a row vector. Variant global features can be used to express and construct a histogram of the entire image. In this paper, we use the following parameters of global feature to extract image features.

1) LBP

Local binary pattern is an effective descriptor for texture, which can measure and extract texture information of an image and keep invariant to light. In this paper, LBP [27] operator is used to retrieve texture information of the entire gray-scale image, and the rotation invariant uniform LBP is used to extract the image texture, where the pixel radius is 2 and 20 adjacent pixels are selected, and we obtain the texture histogram dimension as 383. 141.

2) COLOR LBP

Since LBP feature is extracted from gray-scale image without considering color information, in this paper, in order to retrieve color texture features we extract color LBP features through three channels of RGB color images. Keep the Color LBP feature parameter be same with the LBP feature parameter, extracting features through three channels of the RGB color image, and features of three channels are concatenated to form a 1149-dimensionality histogram feature.

3) GIST

GIST [28] is a global descriptor that captures the spatial structure of an image, which can extract rough but concise contextual information in the way did by human's vision. In this paper, we use the GIST descriptor to retrieve the

spatial layout of the entire image. In the paper, each image is scaled to 128×128 and each Gabor-like filter output to a grid of 4×4 , and the histogram feature dimensionality obtained finally is 512.

4) COLOR GIST

Similar to Color LBP, in this paper we concatenate features of three channels to form a 1536-dimensionality histogram feature by computing GIST features of R, G, and B channels.

5) PHOG

PHOG [29] descriptor can capture the local feature and spatial layout of the image. In this paper we use the descriptor to dig out relations between local objects of an image and its holistic spatial layout. In the experiment, the PHOG feature uses standard parameter settings, dividing the orientation of 0 to 360 degrees into 20 direction angles and finally a 1700-dimensionality histogram feature is obtained.

6) COLOR PHOG

In this paper, PHOG features of R, G, and B channels are computed and concatenated together to form a 5100-dimensionality histogram feature.

7) CIE COLOR SPACE HISTOGRAM

As painters' understanding and application of colors differs greatly, given the disperse color space defined by some color axis, we can obtain a color histogram by calculating the number of occurrences of each disperse color after they are dispersed. The reason why we choose CIE histogram is that the human eye has a more even perception for CIE color space [30]. In this paper, we separate three channels of CIE to 4 parts, and obtain a 64 ($4 \times 4 \times 4$) histogram feature.

8) CANNY EDGE

Shape is the important and powerful attribute of image retrieval, which shows the spatial information that does not exist in the color and texture histogram. In this paper, the shape information of the image is described by the lines of the oil painting and the Canny operator is used to extract the canny edge of oil paintings. The histogram of the edge direction is used to represent the global information about shape attributes of each image, where 0 to 360 degrees of line direction is divided into 30 parts equally, and finally we obtain a 30-dimensionality histogram of line direction.

With the combination of eight global features mentioned above and different kernel functions, we can construct several kernel matrixes of global features for oil painting.

B. EXTRACTION OF LOCAL FEATURE

In this paper, popular Bag-of-Features framework is used to deal with local features. Bag-of-Features-based image shows success when applied in object detection and scene categorization. Imitating the method of Bag-of-Words used in the field of text retrieval, the model describes each image as the unordered set of features extracted from patches/key points.

Using cluster algorithms (eg. K-means) to cluster local features, each cluster center is viewed as a visual word in the vocabulary, like the word in text retrieval. Visual word is represented by code word formed by the corresponding feature of a cluster center (which can be seen as quantization of features). All visual words form a visual vocabulary, corresponding to a code book, that is, a set of code words, and the number of words contained in the vocabulary shows the size of vocabulary. Each feature of the image will be mapped to a word in the Visual Vocabulary. Such mapping can be done by calculating distances between features. Then we count the number of occurrences of each visual word and describe the image as a histogram vector with the same dimension, namely, Bag of Features.

1) COMPLETE LBP

In this paper, Complete LBP [31] is used to extract texture descriptor from the framework of Bag-of-Features. In Complete LBP, a local region is represented by its center pixel and a local difference sign-magnitude transform (LDSMT). In the experiment, feature parameter of Complete LBP keeps consistent with that in the experiment of literature [31].

2) SIFT

In order to capture the appearance of an image, the popular SIFT descriptor is used in this article. The SIFT [32] descriptor performs excellently in tasks of object recognition, texture recognition and motion recognition. In this paper, SIFT information of a grayscale image is extracted and a 128-dimensionality vector is obtained in the end.

3) COLOR SIFT

In order to integrate the color information of the image, the Color SIFT [33] feature for the R, G, and B channels of an image is extracted in this paper. In experiments, OpponentSIFT and CSIFT are also used to matrixes color SIFT features.

4) SSIM

Unlike the descriptors described above, SSIM [34] is a self-similarity descriptor. This feature can measure the layout of an image. The size of each local image is $5 * 5$, and the radius of the relevant area is 40 pixels. And finally we get a 30-dimensionality vector.

With the four local features mentioned above and different kernel functions, several kernel matrixes of local features can be constructed.

IV. CLUSTER MULTIPLE KERNEL LEARNING ALGORITHM

Multiple kernel learning (MKL) algorithm outperforms single kernel learning algorithm in classification. However, all MKL algorithms are burdened with complex calculation. Therefore, reducing the computational complexity of MKL has been the research focus in the field of MKL. CMKL proposes a pre-learning process before the MKL is

carried out, during which the similar candidate sub-kernel is removed and the sub-kernel with better classification ability is selected out. Thus, the size of sub-kernels is reduced and the purpose of unburdening the computational complexity of MKL is achieved. So the main idea of CMKL is as follows: first Combine the input oil painting features with some kernel functions to form a candidate feature sub-core, then update the nuclear clustering center by calculating the similarity of each two sub-cores and the classification performance of the same type of sub-nuclear set. After several iterations, sub-cores with better classification performance can be selected, and feature redundant sub-cores are excluded.

Multiple kernel learning under Lp-norms constraints is as follows:

$$\begin{aligned} \min_{f,b,\xi,d} & \frac{1}{2} \sum_m \frac{1}{d_m} \|f_m\|_{H_m}^2 + C \sum_i \xi_i \\ \text{s.t. } & y_i (\sum_m f_m(x_i) + b) \geq 1 - \xi_i, \quad \forall i \\ & \xi_i \geq 0, \quad \forall i \\ & (\sum_m d_m^p)^{\frac{1}{p}} \leq 1, \quad d_m \geq 0, p \geq 1, \forall m \end{aligned} \quad (1)$$

where H_m is the m th Reproducing Kernel Hilbert Space (RKHS); f_m is the mapping function, which projects the original low dimensional data to high dimensional RKHS. C is equilibrium factor to balance training error; b is the offset; ξ_i is the slack variable to improve the tolerance of training error. d_m is the weight satisfying $\|f\|_M = \sum_m \frac{\|f_m\|_{H_m}^2}{d_m}$, y_i is the label of the i th sample, then the dual formulation of (1) arrives at

$$\begin{aligned} \min_{d_m} \max_{\alpha} & -\frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \sum_m d_m K_m(x_i, x_j) + \sum_i \alpha_i \\ \text{s.t. } & \sum_i \alpha_i y_i = 0, \quad \forall i \\ & C \geq \alpha_i \geq 0, \quad \forall i \\ & (\sum_m d_m^p)^{\frac{1}{p}} \leq 1, \quad d_m \geq 0, p \geq 1, \forall m \end{aligned} \quad (2)$$

Compare to (1), the number of variables to seek solution in (2) reduces from 4 to 2, and the remaining parameter is d_m , which is the kernel weight corresponding to the dual formulation; at the same time, support vector parameter is newly added to satisfy the constraint of $\sum_i \alpha_i y_i = 0$ and $C \geq \alpha_i \geq 0$. α is a Lagrange multiplier. (2) shows that each sub-classifier enjoys the same support vector coefficient, leading to that each classifier cannot perform its own classification ability. Therefore, MKL model can be regarded as the results produced by ensemble learning of several weak classifiers. In this paper, we consider using the approach of judging weak classifiers by ensemble learning to judge the difference of kernel functions.

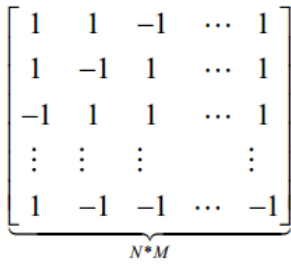


FIGURE 1. View of Oracle output matrix.

A. MEASURE THE SIMILARITY OF TWO KERNEL FUNCTIONS

Ensemble learning is a process that integrating results of several weak classifiers to form a strong classifier. In ensemble learning, the Oracle matrix [32], [33] is constructed to assess the difference and classification ability of weak classifiers. As shown in Fig. 1, assuming there are N training samples and M weak classifiers, then the Oracle matrix is a matrix of $N * M$, where the (n, m) th element represent the result of the n th sample being predicted by the m th weak classifier.

In the pre-training process of CMKL algorithm, leave-one-out method and voting strategy are used to build Oracle output matrix. In the pre-training process, a test sample is reserved every time and the other samples are used as training samples. Since each element in the kernel matrix stands for the inner product of the two samples in high dimensional space, the value of each element in the kernel matrix represents the similarity of the two samples in the high dimensional space. By calculating the similarity between positive test samples and between negative test samples, we can classify the test samples to the category with high similarity. Since each element on the diagonal of kernel matrix stands for the similarity between the test sample and itself, the similarity between the i th and j th sample should be $K(j, i) - K(i, i)$. The similarity between the i th sample and positive samples is

$$Sim_{pos} = \sum_{j=1}^N (K_m(j, i) - K_m(i, i)) * 1\{y_j = 1\} \quad (3)$$

The similarity between the i th sample and negative samples is

$$Sim_{neg} = \sum_{j=1}^N (K_m(j, i) - K_m(i, i)) * 1\{y_j = -1\} \quad (4)$$

Considering that it is maybe unbalance between the number of positive and negative samples, cost penalty factor is used in this paper to solve such unbalance distribution. Thus, the similarity between the i th sample and the positive sample is

$$Sim_{pos} = \frac{\sum_{j=1}^N (K_m(j, i) - K_m(i, i)) * 1\{y_j = 1\}}{N_{pos} - 1\{y_i = 1\}} \quad (5)$$

And the similarity between the i -th sample and the negative sample is

$$Sim_{neg} = \frac{\sum_{j=1}^N (K_m(j, i) - K_m(i, i)) * 1\{y_j = -1\}}{N_{neg} - 1\{y_i = -1\}} \quad (6)$$

In the end, predict sample i belonging to the category with the largest value of similarity. The formulation for the i th sample being predicted by the m th kernel function is shown below:

$$g_m(i) = \text{sign}(Sim_{pos} - Sim_{neg}) \quad (7)$$

If the predicted sample label is exactly the same as the sample, then the corresponding position of the predicted sample in the Oracle matrix is 1, otherwise -1. Distances between every two candidate kernels K_a and K_b is

$$Dis_{a,b} = N - \sum_{i=1}^N g_a(i)g_b(i) \quad (8)$$

$s.t. a < b, \quad a, b \in M$

where the $\sum_{i=1}^N g(i, a)g(i, b)$ is the cosine similarity of candidate kernels K_a and K_b .

B. MEASURE THE CLASSIFICATION ABILITY OF SUB-KERNELS

Kernel permutation is an important approach to evaluate the validity of kernel matrix classification. The value of kernel permutation for a kernel matrix m is obtained by computing the cosine similarity between the kernel matrix m and the ideal kernel ($K_{ideal} = yy^T$).

$$KA(K_m, K_{ideal}) = \langle K_m^*, K_{ideal}^* \rangle_F$$

$$s.t. K_m^* = \frac{K_m}{\sqrt{\langle K_m, K_m \rangle_F}} \quad (9)$$

$$K_{ideal}^* = \frac{K_{ideal}}{\sqrt{\langle K_{ideal}, K_{ideal} \rangle_F}}$$

where the K_m^* is the kernel matrix after regularization, for which the purpose is to put kernel similarity $KA(K_m, K_{ideal})$ in $[-1, +1]$. $\langle K_a, K_b \rangle_F$ is Frobenious distance

$$\langle K_a, K_b \rangle_F = \text{tr}(K_a, K_b) = \sum_{i,j} K_a(x_i, x_j)K_b(x_i, x_j).$$

Therefore, the classification ability of sub-kernel K_m is

$$\beta_m = \langle K_m^*, K_{ideal}^* \rangle_F = \langle K_m^*, y^T y \rangle_F = y K_m^* y^T = g_m y^T \quad (10)$$

$s.t. m \in M$

where K_m^* is the form of K_m after regularization, which contains 3 steps:

(1) regularization of similarity between samples

$$K_m^1 = K_m - A^T$$

$$s.t. A = \text{diag}(K_m) \times \text{ones}(N, N) \quad (11)$$

where $\text{diag}(K_m)$ is the diagonal matrix of K_m and $\text{ones}(N, N)$ is the all one matrix of $N * N$.

(2) regularization of cost factor

$$K_m^2 = B * K_m^1$$

$$s.t. B(i, j) = \frac{1\{y_j = 1\}}{N_{pos} - 1\{y_i = 1\}} - \frac{1\{y_j = -1\}}{N_{neg} - 1\{y_i = -1\}} \quad (12)$$

where “ $*$ ” is dot product sign, representing for the Hadamard product of the two matrixes. The regularization here is used to prevent the unbalance of positive and negative samples.

(3) unit regularization

$$K_m^*(:, j) = \frac{K_m^2(:, j)}{\|K_m^2(:, j)y^T\|}$$

$$s.t. i \in [1, N] \quad (13)$$

In (13), $K_m^*(:, j)$ means the j th column of K_m^* . This regularization is equivalent to the operation of sign function in (7).

After three steps of regularization above, results of classification prediction ability between two kernels can be obtained.

C. CLUSTER MULTIPLE KERNEL LEARNING

Cluster Multiple Kernel Learning (CMKL) is divided into three steps:

(1) use K-medoids to cluster candidate sub-kernels as K-category;

(2) In each category, compute the classification ability of each kernel to select the sub-kernel with best classification performance;

(3) use the selected sub-kernel to carry out multiple kernel learning under $L_p(p > 1)$ -norms constraints.

Algorithm 1 describes the pseudocode of the CMKL algorithm. Algorithm 2 describes the method of computing kernel similarity. The distance calculation and update clustering centers in k- median clustering are described in algorithm 3 and algorithm 4 respectively. Algorithm 5 describes the method of selecting kernels with better classification ability.

Assuming there are f features in data set, k candidate kernel functions, C parameters for each kernel function, and R candidate values for each parameter, then the number of candidate sub-kernels to be optimized is $F \times K \times R^C$. If there are N samples, the minimal value of time complexity to build a sub-kernel is $O(N^2)$ (such as polynomial kernel functions), and space complexity $O(F \times K \times R^C \times N^2)$, then the time complexity to build all sub-kernels is $O(F \times K \times R^C \times N^2)$, the space complexity $O(F \times K \times R^C \times N^2)$, and reducing the number of sub-kernels, namely reduce k , can largely lower the time complexity of building all sub-kernels.

V. EXPERIMENTS

A. DESIGN OF EXPERIMENT

From the analysis of section 4, the advantage of CMKL is that not obviously reducing the accuracy of classification and shortening the time of training classifier. Since the traditional MKL algorithm and the CMKL algorithm can only solve the dichotomous classification at first, in this paper we compare the performance of these algorithms on the UCI dichotomous

Algorithm 1 CMKL

Input:

$X = \{(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i), \dots, (x_n, y_n)\}$: Training set; N : Number of samples;

$K = \{k_1, k_2, \dots, k_m\}$: Candidate kernel matrix set;

M : Number of kernel matrix; p_norm : Value of L_p norm constraint; C : Number of cluster centers; T : Number of iterations; VC : Cluster center vector;

Output: $h(x_i)$: predict label for x_i

$G = \text{KernelSimilarity}(X, K, M, N)$

$KC = \{k_{c1}, k_{c2}, \dots, k_{cc}\} \leftarrow \text{RandomSelectKernel}(C)$

3. $VC = \text{RandomClusterCenter}(VC)$

3. for $t=1$ to T do

4. $SC \leftarrow \phi$

5. $SC = \text{KernelDistances}(SC, G, N, C, VC)$

6. $VC = \text{UpdateClusterCenter}(SC, C, M)$

7. $\text{PickKernel} = \text{SelectKernels}()$

8. end for

9. for $i=1$ to N do

10. $h(x_i) = \text{MKL}(\text{PickKernel}, p_norm)$

11. end for

return $h(x_i)$

Algorithm 2 KernelSimilarity

Input:

$X = \{(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i), \dots, (x_n, y_n)\}$: Training set; N : Number of samples;

$K = \{k_1, k_2, \dots, k_m\}$: Candidate kernel matrix set;

M : Number of kernel matrix;

Output: G : Kernel similarity matrix;

1. for $m=1$ to M do

2. for $i=1$ to N do

$$3. \quad Sim_{pos}^{m,i} = \frac{\sum_{j=1}^N (K_m(j, i) - K_m(i, i)) * 1\{y_j = 1\}}{N_{pos} - 1\{y_i = 1\}}$$

$$4. \quad Sim_{neg}^{m,i} = \sum_{j=1}^N (K_m(j, i) - K_m(i, i)) * 1\{y_i = -1\}$$

$$5. \quad g_m(i) = \text{sign}(Sim_{pos}^{m,i} - Sim_{neg}^{m,i})$$

6. end for

7. end for

8. $G = [g_1, g_2, \dots, g_M]$

return G

data sets and then conduct the multi-class experiment on the Painting 91 data set.

B. SPECIFICATION OF DATA SETS

1) UCI DATA SETS

From the UCI data sets, the paper chooses four data sets: breast, bupa, pima and wdbc. 80% samples are selected randomly for training and the remaining 20% for testing.

Algorithm 3 KernelDistances

Input: SC: The categories of each candidate kernel;
 G: Kernel similarity matrix; N: Number of samples;
 C: Number of cluster centers; VC: Cluster center vector;
 Output: SC: The categories of each candidate kernel;

```

1. for m=1 to N do
2.   for c=1 to C do
3.      $Dis_{m,vc_c} = N - \sum_{i=1}^N g_m(i)g_{vc_c}(i)$ 
4.   end for
5.    $SC \leftarrow SC \cup \{\arg \min(Dis_m)\}$ 
6. end for
return SC
```

Algorithm 4 UpdateClusterCenter

Input: SC: The categories of each candidate kernel;
 C: Number of cluster centers; M: Number of kernel matrix;
 Output: VC: Cluster center vector;

```

1. for c=1 to C do
2.   for m1=1 to M do
3.     if  $SC_{m1} = c$  then  $J_c = 0$ 
4.     for m2=1 to M do
5.       if  $SC_{m2} = c$  then
          $J_c = J_c + Dis_{m1,m2}$ 
6.     end if
7.   end for
8. end if
9. end for
10.  $vc_c = \arg \min(J_c)$ 
11. end for
return VC
```

2) PAINTING91 DATA SET

Painting91 data set contains 2275 training samples and 1991 test samples from 91 painters in total.

C. MOBILE APPLICATION PROCESS OF OIL PAINTER RECOGNITION

In this paper, we create a mobile phone application that uses the camera of the mobile phone to obtain the oil painting, and then uploads this to the server. The feature extraction and recognition of the oil painting would be completed on the server, and the recognition result would be returned to the mobile phone. In the mobile phone application, the image data can be sent to the server by uploading pictures in the mobile phone album or the image captured by the camera of the mobile phone in real time. We can also manually edit the image before uploading, such as cropping, rotating and some other machine learning methods [39]–[43] on the preview page. This would improve the quality of uploaded images and improve recognition efficiency.

Algorithm 5 SelectKernels

Input: $K = \{k_1, k_2, \dots, k_m\}$: Candidate kernel matrix set;
 M: Number of kernel matrix; C: Number of cluster centers;
 SC: The categories of each candidate kernel;
 Output: PickKernel: The kernels which have the best classification ability;

```

1. PickKernel  $\leftarrow \phi$ 
2. for c=1 to C do
3.    $\beta_m \leftarrow \phi$ 
4.   for m=1 to M do
5.     if  $SC_m = c$  then  $\beta_m \leftarrow \beta_m \cup \{g_m y^T\}$ 
6.   end if
7. end for
8. PickKernelIndex  $\leftarrow \arg \min(\beta_m)$ 
9. PickKernel  $\leftarrow \text{PickKernel} \cup \{k_{\text{PickKernelIndex}}\}$ 
10. end for
return PickKernel
```

TABLE 1. Specification of UCI data sets.

Data Sets	Training Set	Test Set	Dimensionality
breast	546	137	9
bupa	276	69	6
pima	614	154	8
wdbc	455	114	31

TABLE 2. Specification of Painting91 dataset.

Data Sets	Painting
Training size of each artist	25
Test size of each artist	7-25
Training set	2275
Test set	1991

D. EXPERIMENT BASED ON UCI DATA SETS

According to the experimental settings in literature [23], the kernel matrix is constructed by using the Gaussian kernel function and the polynomial kernel function, where the 10 candidate parameters of Gaussian kernel function are [0.5,1,2,5,7,10,12,15,17,20] and 3 candidate parameters of polynomial kernel function are [1]–[3]. Since the dimensionality of the samples in the four data sets is relatively low, it is not only necessary to use each dimension of data alone to build a sub-kernel, but also all dimensions of data to build a sub-kernel. Assuming that the dimension of data set is M, and each data set can produce candidate sub-kernels at the number $(M + 1) \times (10 + 3)$. In the experiment, in order to ensure that each sub-kernel is positive definite, the approach in literature [23] is taken to regularize all sub kernels by matrices. Since the results of clustering algorithm are influenced by the initial value greatly, CMKL need to operate 20 times on each data set to obtain

TABLE 3. Number of clusters in different UCI data sets.

Data Sets	Number of clusters
Training size of each artist	25
Test size of each artist	7-25
Training set	2275
Test set	1991

TABLE 4. Classification accuracy of different algorithms on UCI data sets.

Data Sets	breast	bupa	pima	wdbc
MKL-L1	0.932	0.605	0.776	0.932
MKL-L2	0.934	0.609	0.773	0.930
CMKL-L2	0.942	0.648	0.778	0.932
MKL-L3	0.920	0.652	0.766	0.921
CMKL-L3	0.932	0.656	0.778	0.937
MKL-L4	0.920	0.652	0.760	0.930
CMKL-L4	0.927	0.661	0.765	0.942
MKL-L5	0.920	0.652	0.779	0.939
CMKL-L5	0.932	0.665	0.784	0.947
MKL-L10	0.934	0.652	0.766	0.947
CMKL-L10	0.953	0.671	0.775	0.955
MKL-L100	0.944	0.667	0.760	0.965
CMKL-L100	0.961	0.680	0.764	0.959
MKL-L ∞	0.944	0.670	0.757	0.968
CMKL-L ∞	0.958	0.678	0.764	0.960
Lpboost	0.965	0.667	0.772	0.971
MKL-elas	0.968	0.675	0.768	0.965

the final average and the box plot. Table 3 shows the number of cluster centers in different datasets. We explored the performance of different algorithms on four UCI datasets in Table 3. Fig. 2 evaluates the accuracy rate of CMKL on four UCI data sets. Table 5 and Fig. 3 evaluate the performance of the algorithm in time consuming and clustering centers respectively.

Table 4 lists the average classification accuracy of traditional MKL algorithm and CMKL algorithm based on UCI four data sets under the different norm constraints. For data breast and bupa, results of MKL with L1-norm constraints is not so accurate as that with Lp($p \geq 2$). The reason maybe that data breast and bupa prefers the non-sparse data, which is unsuitable for L1 sparsity constraints. In the case of same norm constraints, CMKL algorithm outperforms any other traditional MKL algorithms in terms of classification accuracy. The reason lies in the pre-training stage, during which a large scale of similar candidate sub-kernels are removed in CMKL, and sub-kernels with better classification performance are selected out. It not only ensures that the selected sub-kernels have greater differences, but also secures the classification accuracy of each sub-kernel, which is the same reason that greater the differences between sub-classifiers, higher the accuracy. No matter the data is sparse or not in structure, CMKL can take the sparseness and accuracy of MKL into account. Linear Programming Boosting (LPboost) enjoys a slightly higher accuracy of classification than CMKL

TABLE 5. Operating time consumed by different algorithms on UCI data sets (second).

Data Sets	breast	bupa	pima	wdbc
MKL-L1	313.50	450.99	618.07	1568.30
MKL-L2	46.77	27.11	130.14	138.92
CMKL-L2	35.49	11.13	86.12	89.40
MKL-L3	55.06	34.47	179.82	155.57
CMKL-L3	37.04	11.19	99.03	84.85
MKL-L4	56.69	41.94	214.81	159.77
CMKL-L4	33.31	16.56	121.13	86.10
MKL-L5	55.29	46.69	249.70	160.23
CMKL-L5	36.88	15.67	184.28	89.92
MKL-L10	55.52	58.80	324.15	170.82
CMKL-L10	33.91	13.43	183.91	92.43
MKL-L100	66.26	70.24	459.38	181.66
CMKL-L100	37.77	18.69	317.20	92.47
MKL-L ∞	4.86	3.57	15.02	15.97
CMKL-L ∞	3.94	1.83	8.55	9.69
Lpboost	2260.34	3698.12	4697.33	8625.65
MKL-elas	47.83	28.52	132.31	140.77

on data sets breast, pima, and wdbc. MKL-elas under elastic network constraints enjoys higher accuracy of classification than MKL-L2 and MKL-L1, as norm constraints are between L1 and L2.

It can be seen from Fig. 2 that the classification accuracy of CMKL algorithm is unstable, and the accuracy of certain classification is lower than that of traditional MKL algorithm. However, the average classification accuracy of CMKL algorithms is higher than that of traditional MKL algorithms, and it enjoys a larger probability of giving better results than the average accuracy.

Table 5 shows the average time consumed in training by different algorithms. Though the MKL algorithm under the L1-norm constraints can produce sparse solution, and its classification accuracy on non-sparse data is higher than that under Lp ($p \geq 2$) norms constraints, the training time of this algorithm is far longer than that under Lp ($p \geq 2$) norms constraints. Although the classification accuracy of LPboost is higher than that of CMKL-Lp, the training time consumed by LPboost is even more 5 times higher than that by MKL-L1. The training time of MKL-elas is between that of MKL-L1 and MKL-L2. The training time of CMKL consists of two parts: pre-training time and time consumed by multiple kernel learning. We can see that the total training time of CMKL is always far shorter than MKL under Lp norms constraints directly, even the pre-training time is added. Results of experiments reflect that, reducing the number of candidate sub-kernels is the key to unburden the complexity of multiple kernel learning.

In the pre-training process, we remove and select candidate sub-kernels, which shortens the training time largely and saves computing sources without affecting classification performance of multiple kernel learning.

Fig. 3 shows the classification accuracy of CMKL-L2 algorithm on four data sets. It is easy to see that, with the increase

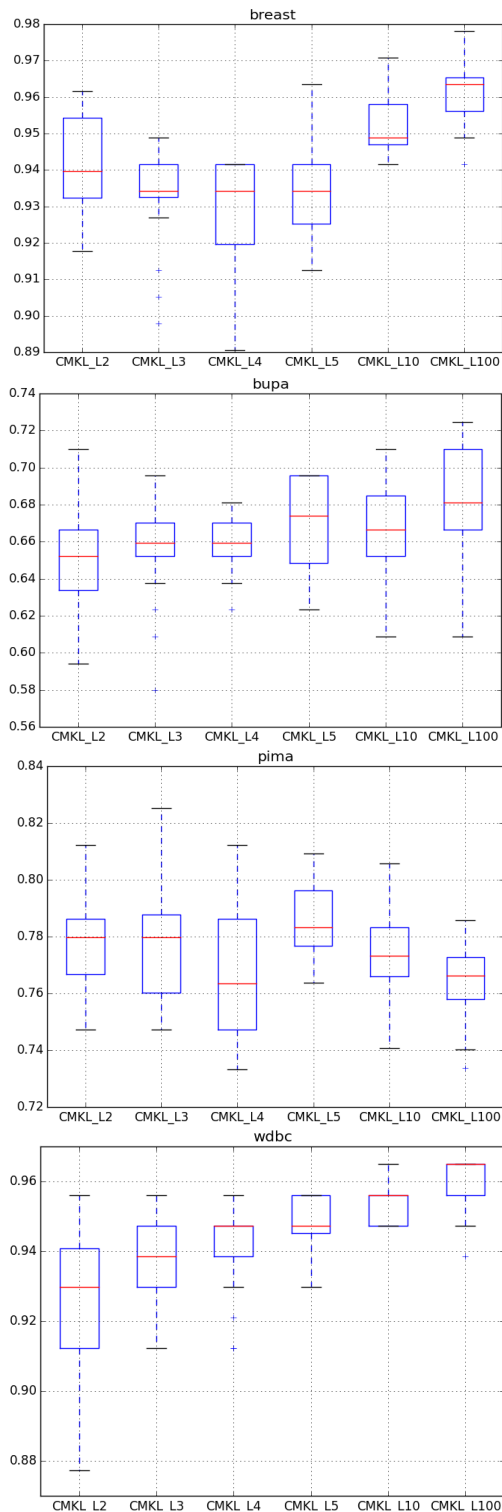


FIGURE 2. The box plot of accuracy rate of CMKL on 4 UCI data sets.

of cluster centers, the accuracy rises progressively first and then falls progressively. When the number of cluster centers is small, though computing resources is saved, useful classification information is few and insufficient fitting of model leads to low classification accuracy. When the number of cluster

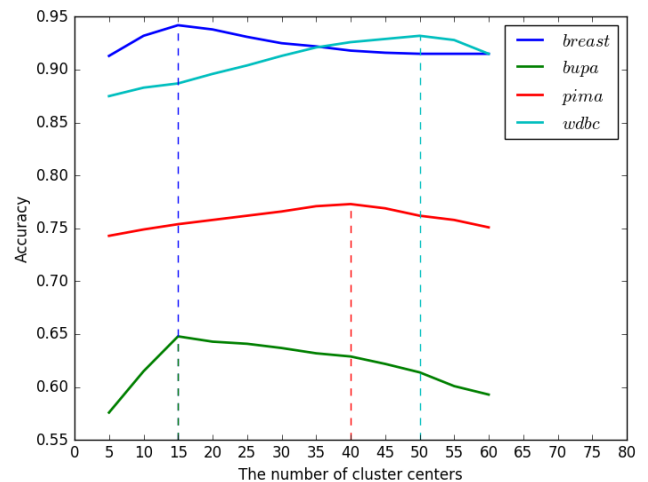


FIGURE 3. Accuracy of CMKL-L2 at different dataset.

centers is large, that is, when candidate sub-kernels increases, the classification accuracy is not improved. The reason is that too many sub-kernels are doped with redundant information which interferes the classification effects. Therefore, selecting the number of cluster centers suitably is very helpful for improving the classification accuracy.

E. EXPERIMENTS BASED ON PAINTING 91 DATA SET

The classifier used in the single-feature classification experiment is the LIBSVM package developed by Professor Lin Zhiren of Taiwan University. The multi-classifier generalize 1vs1 rule to dichotomous SVM classifier. The parameter of multiclass SVM classifier is selected by 5 fold cross validation. In the experiment of multi-feature fusion, 1 vs 1 rule is also used to generalize dichotomous classifier of MKL to multiclass classifier so as to carry out classification experiments on painting 91 data set. Candidate features are set with reference to the settings in literature [8] and [13]. 10 candidate parameters of Gaussian kernel function is [0.5,1,2,5,7,10,12,15,17,20]. 3 candidate parameters of polynomial kernel function is [1]–[3]. With reference to literature [40], chi-square distance between samples is used in place of Euclidean distance to construct chi-square kernel matrix. Candidate parameter of chi-square kernel function is same with that of Gaussian kernel function. The number of clusters of CMKL is 30. In this paper, we construct 322(14*23) kernel matrices respectively for 8 global features and 6 local features.

In Table 6, we evaluate the accuracy of single feature on oil painters recognition. Table 7 and Table 8 show the accuracy and computation time of different algorithms on Painting91 data set. Fig. 4 is a graphical representation corresponding to Table 7. Fig. 5 shows the accuracy of CMKL-L2 at various number of cluster centers. The top5 classification results of CMKL in the Painting91 dataset is described in Fig. 6.

Table 6 records the classification accuracy of different single features in the experiments of oil painter recognition.

TABLE 6. Classification accuracy of single feature on Painting 91 data set.

Name of features	Accuracy
LBP	0.285
Color LBP	0.350
PHOG	0.186
Color PHOG	0.228
GIST	0.239
Color GIST	0.278
CIE histogram	0.252
Canny edge	0.316
SIFT	0.426
RGB SIFT	0.403
Complete LBP	0.347
CSIFT	0.364
Opponent SIFT	0.395
SSIM	0.237

TABLE 7. Classification accuracy of different algorithms on Painting91 data set.

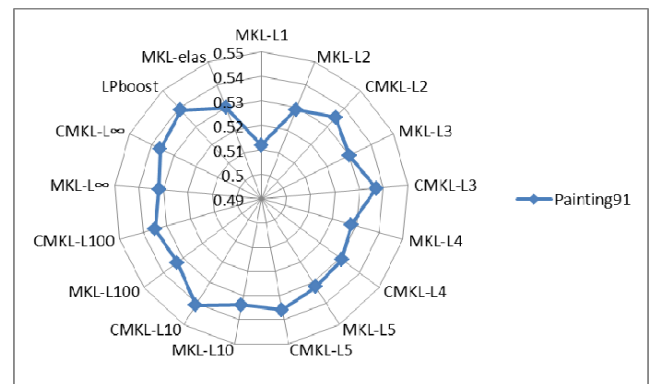
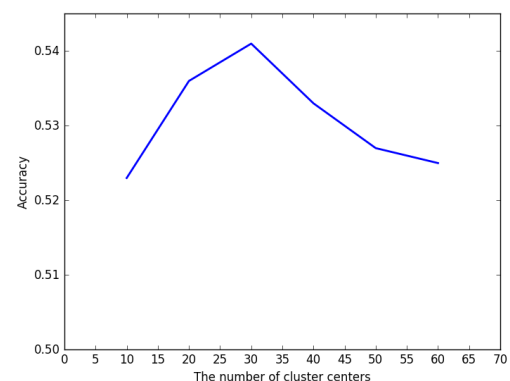
Algorithm	Accuracy
MKL-L1	0.512
MKL-L2	0.529
CMKL-L2	0.535
MKL-L3	0.530
CMKL-L3	0.537
MKL-L4	0.528
CMKL-L4	0.531
MKL-L5	0.532
CMKL-L5	0.536
MKL-L10	0.534
CMKL-L10	0.545
MKL-L100	0.533
CMKL-L100	0.535
MKL-L ∞	0.532
CMKL-L ∞	0.536
Lpboost	0.539
MKL-elas	0.530

The accuracy rates of Color LBP, Color GIST, and Color PHOG are higher than the corresponding LBP, GIST and PHOG, which means color information is important in reflecting an oil painter's artistic style. RGB SIFT is best at representing among three Color SIFT descriptors. Canny edge and SSIM can also narrate artistic styles of oil painters very well. Features above always describe the artistic style of oil painters in one respect, so reasonable fusion of multiple features may provide a more comprehensive description.

Table 7 records the classification results of the multi-feature fusion on the Painting91 data set for different algorithms. It can be seen that, compared to the classification results of MKL under L1 norm constraints, better results are obtained by MKL under Lp($p \geq 2$) norms constraints. Lpboost obtains higher accuracy than MKL-Lp, but at the cost of consuming much more training time. The classification accuracy of MKL-elas is higher than that of MKL-L1 and MKL-L2, and the length of training time of MKL-elas is

TABLE 8. Operating time consumed by different algorithms on Painting91 data sets (sec).

Algorithm	Training time
MKL-L2	58.30
CMKL-L2	37.41
MKL-L3	60.43
CMKL-L3	37.90
MKL-L4	59.39
CMKL-L4	36.35
MKL-L5	60.15
CMKL-L5	36.47
MKL-L10	63.42
CMKL-L10	38.52
MKL-L100	69.67
CMKL-L100	37.62
MKL-L ∞	6.02
CMKL-L ∞	3.97
Lpboost	402.53
MKL-elas	60.19

**FIGURE 4.** Classification accuracy of different algorithms on Painting91 data set.**FIGURE 5.** Accuracy of CMKL-L2 at various number of cluster centers.

between that of MKL-L1 and MKL-L2. CMKL outperforms MKL under Lp norms constraints in aspects of classification accuracy, and when p is 10 it achieves the highest accuracy of classification. Table 8 records the average training time of a dichotomous classifier (in the experiment, it is the average training time of $91 * (91 - 1) / 2 = 4095$ dichotomous classifiers). From Table 8 we can see that CMKL algorithm

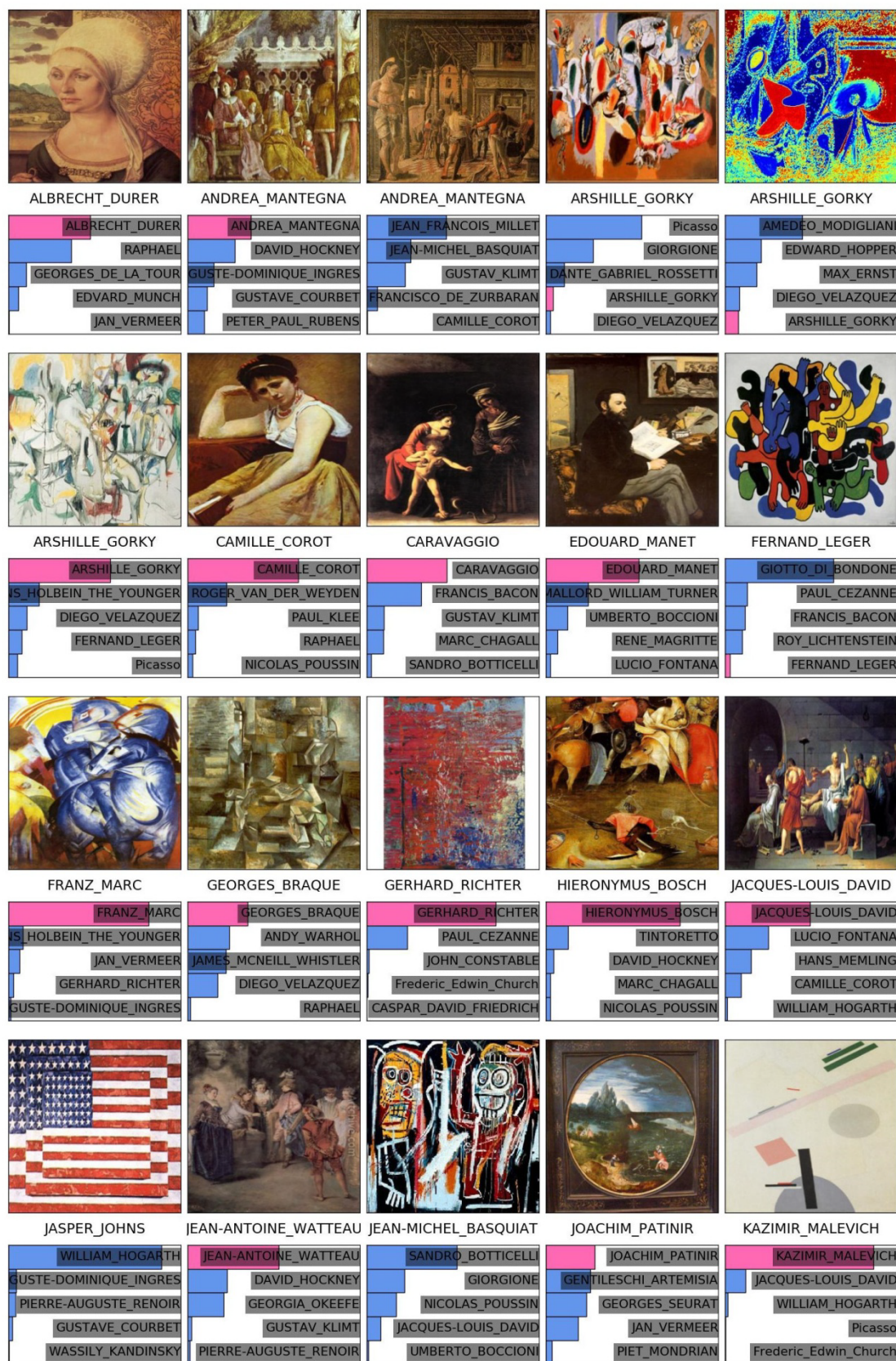


FIGURE 6. The top5 classification results of CMKL in the Painting91 dataset.

has great advantage over others when there are a plenty of candidate sub-kernels. It proves that pre-training process is vitally important for unburden the complexity of MKL computation. Removing and selecting candidate sub-kernels in a rational way contributes to the improvement of the classification accuracy. For a large scale of oil painters recognition the rapid CMKL algorithm is a good choice. From Fig. 6 we can see that when the number of cluster centers is 30, the model achieves best results of classification on Painting91 data set. Therefore, in the experiment the number of cluster centers is set at 30.

VI. CONCLUSION AND PROSPECT

This paper proposes a novel and rapid Cluster Multiple Kernel Learning algorithm to recognize oil painters. Although multiple kernel learning under L_p norms constraints displays better classification performance than single kernel learning in the aspects of multi-feature fusion, the computing resources it consumed is rather large. Through a pre-training process CMKL algorithm excludes and selects candidate sub-kernels with great difference and better classification ability, and then the selected sub-kernels are used in multiple kernel learning under L_p ($p \geq 2$) norms constraints. CMKL outperforms MKL under L_p norms constraints in aspects of classification accuracy and training time.

There are many problems to be solved in the field of recognition of oil painters. Firstly, whether there are more suitable features for the selection of manual features? Secondly, when carrying out multi-feature fusion and classification, except multiple kernel learning, if the neural network can be considered to learn the classification results of single feature and then conducting the feature confusion? At the same time, the CMKL proposed in this paper has many problems to be solved. For example, when generalizing 1vs1 rule, whether the number of clusters can change dynamically? How to select a more suitable clustering initial value of kernel matrix? If there are some relations between the number of clusters and sample structures? And we can use more visual technologies [38] to present the results.

REFERENCES

- [1] B. Saleh and A. Elgammal, "Large-scale classification of fine-art paintings: Learning the right metric on the right feature," *J. Comput. Sci. Technol.*, vol. 25, no. 3, pp. 595–605, 2015.
- [2] Y. Bar, N. Levy, and L. Wolf, "Classification of artistic styles using binarized features derived from a deep neural network," in *Proc. Eur. Conf. Comput. Vis.*, Zürich, Switzerland, Sep. 2014, pp. 71–84.
- [3] A. Rakotomamonjy, X. Mary, and S. Canu, "Non-parametric regression with wavelet kernels," *Appl. Stochastic Models Bus. Ind.*, vol. 21, no. 2, pp. 153–163, 2010.
- [4] I. Widjaja, W. Leow, and F. Wu, "Identifying painters from color profiles of skin patches in painting images," in *Proc. Int. Conf. Image Process.*, 2003, vol. 1, no. 1, pp. 845–848.
- [5] S. Lyu, D. Rockmore, and H. Farid, "A digital technique for art authentication," *Proc. Nat. Acad. Sci. USA*, vol. 101, no. 49, pp. 17006–17010, 2004.
- [6] C. R. Johnson et al., "Image processing for artist identification," *IEEE Signal Process. Mag.*, vol. 25, no. 4, pp. 37–48, Jul. 2008.
- [7] J. Shen, "Stochastic modeling western paintings for effective classification," *Pattern Recognit.*, vol. 42, no. 2, pp. 293–301, 2009.
- [8] J. Zujovic, L. Gandy, S. Friedman, B. Pardo, and T. Pappas, "Classifying paintings by artistic genre: An analysis of features & classifiers," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, Oct. 2009, pp. 1–5.
- [9] B. Siddiquie, S. N. Vitaladevuni, and L. S. Davis, "Combining multiple kernels for efficient image classification," in *Proc. Workshop Appl. Comput. Vis.*, 2009, pp. 1–8.
- [10] L. Shamir, T. Macura, N. Orlov, D. M. Eckley, and I. G. Goldberg, "Impressionism, expressionism, surrealism: Automated recognition of painters and schools of art," *ACM Trans. Appl. Perception*, vol. 7, no. 7, pp. 177–185, 2010.
- [11] G. Carneiro, N. P. da Silva, A. Del Bue, and J. P. Costeira, "Artistic image classification: An analysis on the PRINTART database," in *Proc. Eur. Conf. Comput. Vis.*, vol. 7575, no. 1, 2012, pp. 143–157.
- [12] F. Khan, S. Beigpour, J. Van de Weijer, and M. Felsberg, "Painting-91: A large scale database for computational painting categorization," *Mach. Vis. Appl.*, vol. 25, no. 6, pp. 1385–1397, 2014.
- [13] L. Gatys, A. Ecker, and M. Bethge. (2015). "A neural algorithm of artistic style." [Online]. Available: <https://arxiv.org/abs/1508.06576>
- [14] K.-C. Peng and T. Chen, "Cross-layer features in convolutional neural networks for generic classification tasks," in *Proc. Int. Conf. Image Process.*, Sep. 2015, pp. 3057–3061.
- [15] K.-C. Peng and T. Chen, "A framework of extracting multi-scale features using multiple convolutional neural networks," in *Proc. Int. Conf. Multimedia Expo*, 2015, pp. 1–6.
- [16] G. Folego, O. Gomes, and A. Rocha, "From impressionism to expressionism: Automatically identifying van Gogh's paintings," in *Proc. Int. Conf. Image Process.*, Sep. 2016, pp. 141–145.
- [17] R. Anwer, F. Khan, V. Joost, and J. Laaksonen, "Combining holistic and part-based deep representations for computational painting categorization," in *Proc. Comput. Vis. Image Process.*, 2016, pp. 339–342.
- [18] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *J. Mach. Learn. Res.*, vol. 5, no. 4, pp. 27–72, 2004.
- [19] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proc. Int. Conf. Mach. Learn.*, vol. 6, 2004, pp. 43–51.
- [20] C. Lemaréchal and C. Sagastizabal, "Practical aspects of the Moreau–Yosida regularization: Theoretical preliminaries," *SIAM J. Optim.*, vol. 7, no. 2, pp. 367–385, 1997.
- [21] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2491–2521, 2008.
- [22] M. Kloft, U. Brefeld, P. Laskov, and S. Sonnenburg, "Non-sparse multiple kernel learning," in *Proc. NIPS Workshop Kernel Learn., Autom. Selection Optim. Kernels*, vol. 7, no. 6, 2008, pp. 775–782.
- [23] M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskov, K. R. Müller, and A. Zien, "Efficient and accurate lp-norm multiple kernel learning," in *Proc. NIPS*, vol. 22, 2009, pp. 997–1005.
- [24] Z. Xu, R. Jin, H. Yang, I. King, and M. R. Lyu, "Simple and efficient multiple kernel learning by group lasso," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 1175–1182.
- [25] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Roy. Statist. Soc., B (Statist. Methodol.)*, vol. 68, no. 1, pp. 49–67, 2006.
- [26] X. Xu, I. W. Tsang, and D. Xu, "Soft margin multiple kernel learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 5, pp. 749–761, May 2013.
- [27] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.
- [28] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.
- [29] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proc. CIVR*, 2007, pp. 401–408.
- [30] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.
- [31] Z. Guo and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1657–1663, Jan. 2010.
- [32] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
- [33] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1582–1596, Sep. 2010.

- [34] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *Proc. CVPR*, Jun. 2007, pp. 1–8.
- [35] E. K. Tang, P. N. Suganthan, and X. Yao, "An analysis of diversity measures," *Mach. Learn.*, vol. 65, no. 1, pp. 247–271, 2006.
- [36] X. Luo et al., "Short-term wind speed forecasting via stacked extreme learning machine with generalized correntropy," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4963–4971, Nov. 2018.
- [37] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *Int. J. Comput. Vis.*, vol. 73, no. 2, pp. 213–218, 2007.
- [38] Z. Liao, D. He, Z. Chen, X. Fan, Y. Zhang, and S. Liu, "Exploring the characteristics of issue-related behaviors in GitHub using visualization techniques," *IEEE Access*, vol. 6, pp. 24003–24015, 2018.
- [39] M. Chen, Y. Li, X. Luo, W. Wang, L. Wang, and W. Zhao, "A novel human activity recognition scheme for smart health using multilayer extreme learning machine," *IEEE Internet Things J.*, to be published. doi: [10.1109/JIOT.2018.2856241](https://doi.org/10.1109/JIOT.2018.2856241).
- [40] X. Luo, X. Yang, C. Jiang, and X. Ban, "Timeliness online regularized extreme learning machine," *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 3, pp. 465–476, 2018.
- [41] X. Luo et al., "Towards enhancing stacked extreme learning machine with sparse autoencoder by correntropy," *J. Franklin Inst.*, vol. 355, no. 4, pp. 1945–1966, Mar. 2018. doi: [10.1016/j.jfranklin.2017.08.014](https://doi.org/10.1016/j.jfranklin.2017.08.014).
- [42] Z. Liao et al., "A prediction model of the project life-span in open source software ecosystem," *Mobile Netw. Appl.*, to be published. doi: [10.1007/s11036-018-0993-3](https://doi.org/10.1007/s11036-018-0993-3).
- [43] J. Zhang, Z. Zhang, Z. Chen, and L. Deng, "Integrating multiple heterogeneous networks for novel lncRNA-disease association inference," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, to be published. doi: [10.1109/TCBB.2017.2701379](https://doi.org/10.1109/TCBB.2017.2701379).



TIAN ZHOU received the B.E. degree in information science and engineering from Central South University, China, in 2016, where currently pursuing the master's degree with the School of Information Science and Engineering. His research interests include image processing and vision processing.



XIAOPING FAN received the Ph.D. degree from the South China University of Technology, China, in 1998. He is currently a Professor with the Hunan University of Finance and Economics, China. His main research interests include data visualization and robotics and open source software social networks.

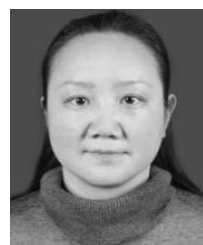


YAN ZHANG received the Ph.D. degree in computer science from Middlesex University, London, U.K. She is currently a Lecturer with the Department of Computer, Communication and Interactive Systems, School of Engineering and Built Environment, Glasgow Caledonian University, Scotland, U.K. Her research interests include data visualization, data analytics, big data, data mining, and tele-health.



JINSONG WU received the Ph.D. degree from Queen's University. He is currently an Associate Professor with the Department of Electrical Engineering, Universidad de Chile, Chile. His research interests include machine learning and green communication.

...



ZHIFANG LIAO was born in Changsha, Hunan, China, in 1968. She received the B.S. degree in industry control engineering and the M.S. degree in computer science from the Changsha Railway Institute, in 1998, and the Ph.D. degree in computer technology and application from Central South University, in 2008.

From 1990 to 1997, she was an Engineer with Hunan Computer Factory. Since 1997, she has been a Lecturer/an Associate Professor with Central South University. Her research interests include open source software, open source software ecosystems, data mining, and so on.



LE GAO received the B.E. degree in software engineering from Xidian University, China, in 2015, and the master's degree in software engineering from Central South University, China, in 2018. His main research interests include data visualization, data mining, and the analysis of user's behavior in the open-source community.